

PB-217 079

**A DIGITAL TECHNIQUE FOR CALCULATING AND
PLOTING DISSOLVED OXYGEN DEFICITS**

**Public Health Service
Charlottesville, Virginia**

1965

DISTRIBUTED BY:

NTIS

**National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151**

FE 67.9
ZXT
00 896

PB 217 079

**A DIGITAL TECHNIQUE FOR CALCULATING
AND PLOTTING DISSOLVED OXYGEN DEFICITS**

Reproduced by
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U S Department of Commerce
Springfield VA 22151

CB-SRBP Technical Paper No. 1
Bunce and Wastler
PHS-DWS&PC
Region III

LIBRARY COPY

Hudson-Champlain Project, USDHEW

21

LIBRARY COPY

TABLE OF CONTENTS

	<u>Page</u>
List of Figures	
I. Introduction	1
II. Oxygen Sag Equation	1
III. Main Program and Subroutines	2
IV. Preparation of Data	5
V. Data Output	8
VI. Cost Data	9
VII. Conclusions	10
VIII. Acknowledgments	10

I

LIBRARY COPY

Hudson-Champlain Project, USDHEW

LIST OF FIGURES

- Figure 1 Simplified flow chart showing operations performed by the program
- Figure 2 Main program for the oxygen sag equation
- Figure 3 Subroutine MAP - used by the main program
- Figure 4 Subroutine WRITEM - used by subroutine MAP
- Figure 5 Data package format for Set No. 1
- Figure 6 Data output and graph for Set No. 1
- Figure 7 Data output and graph for Set No. 2
- Figure 8 Data output and graph for Set No. 3

I. Introduction

A package program for use on the IBM 7090 computer is presented for obtaining a numerical solution of the classical Streeter-Phelps oxygen sag equation. Oxygen deficits are computed for each desired time and plotted as a function of time.

In general, no knowledge on the reader's part is assumed with regard to computers or the program presented. Certain terminology having general use will be discussed in order to clarify the presentation; however, the step-by-step instructions should enable engineers and scientists to obtain correct results without a thorough knowledge of computer terminology. ()

II. Oxygen Sag Equation

In 1925 Streeter-Phelps⁽¹⁾ formulated the oxygen sag equation which describes the changes in dissolved oxygen content of a stream following the introduction of an oxygen demanding organic waste. The mathematical formulation describes the net rate of change in the oxygen deficit (D) at any time (t) and is expressed as a first order differential equation:

$$\frac{dD}{dt} = K_1L - K_2D \quad \text{Equation (1)}$$

where K_1L represents the increase in the deficit at a rate which is assumed to be proportional to the oxygen demand of the waste. The second term of the equation, namely $-K_2D$, represents the rate of reaeration expressed in terms of the oxygen saturation deficit.

On integrating Equation (1) the dissolved oxygen content of the water may be obtained as the deficit below saturation at any time. The integrated form is then:

$$D = \frac{K_1 L_a}{K_2 - K_1} (e^{-K_1 t} - e^{-K_2 t}) + D_a e^{-K_2 t} \quad \text{Equation (2)}$$

where: D_a = Initial dissolved oxygen saturation deficit (mg/l)

D = Oxygen deficit, after time (t) (mg/l)

L_a = Ultimate biochemical oxygen demand (mg/l)

K_1 = Empirical deoxygenation rate constant

K_2 = Empirical reaeration rate constant

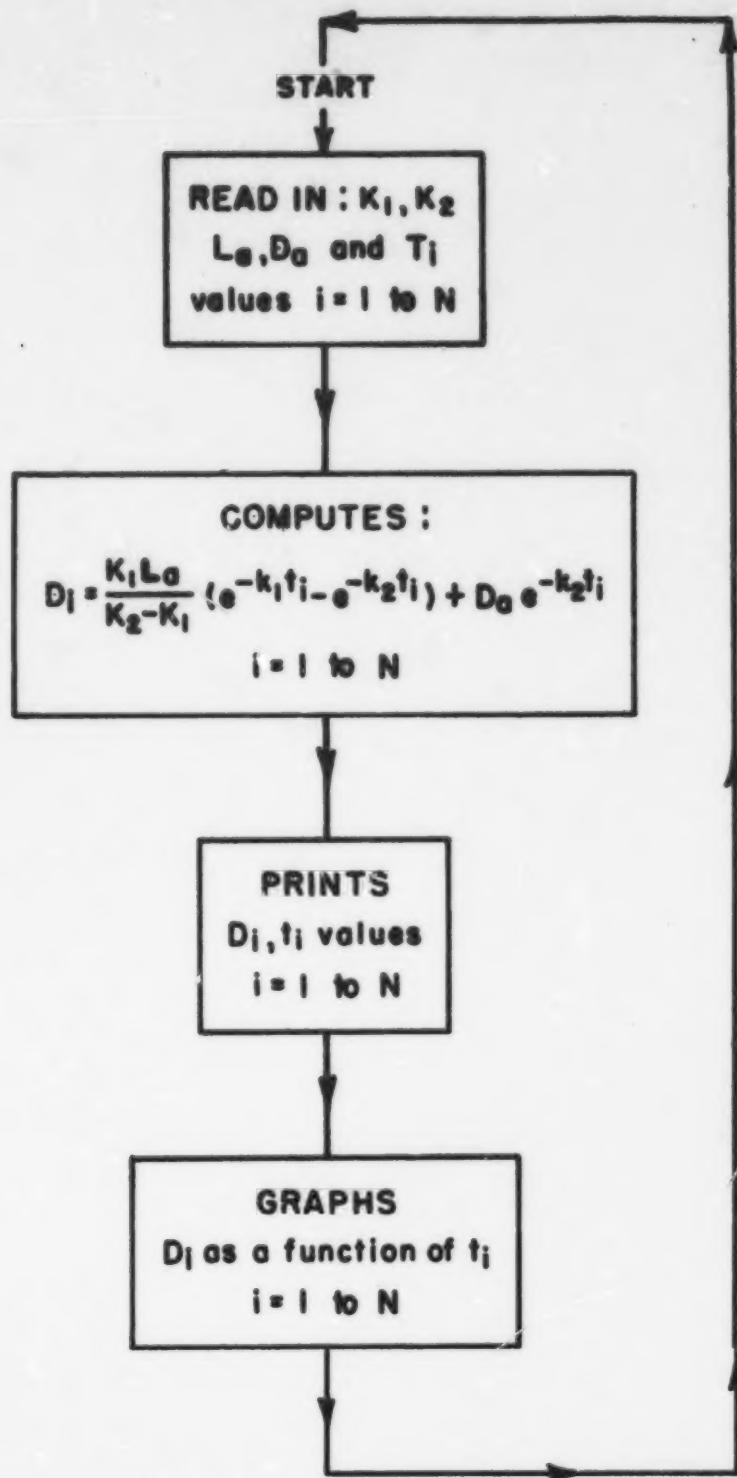
t = Elapsed time in days

e = Base of naperian or natural logarithms and is equal to 2.71828

The numerical solution to the above form requires that knowledge of the reaeration and deoxygenation rate constants together with the ultimate biochemical oxygen demand and initial oxygen deficit be known. Once these values are in hand, we may solve for the oxygen deficits at any desired time.

III. Main Program and Subroutines

In order to visualize the flow of data and functions of the program, Figure 1 is presented showing a simplified flow chart of the various operations performed. The input information is seen to consist of K_1 , K_2 , L_a , D_a , and T_i values where $i = 1$ to N . The program computes the corresponding D_i value or each t_i value for a given K_1 , K_2 , L_a , and D_a . The computed



SIMPLIFIED FLOW CHART SHOWING OPERATIONS
PERFORMED BY THE PROGRAM

3a

FIGURE 1

t_1 and D_1 values for each set of input data are printed and graphed. After the first set of data has been computed, control returns to the beginning of the program and new values for K_1 , K_2 , L_a , D_a , and t_1 's are read in. This process continues until the last set of data has been computed and printed.

The program presented here has been written in what is termed FORTRAN. Every type of computer is designed to respond to a special code, called a "machine language." The instructions telling the computer what steps to perform to solve a problem must be given to the computer in its own language. The FORTRAN program makes it unnecessary for the programmer to learn the more complicated machine language. The word FORTRAN, derived from formula translation, is a method of programming using language similar to familiar usage. The FORTRAN program is then assembled by the computer in proper form for machine interpretation.

The main program for calculating dissolved oxygen deficits and plotting the deficits as a function of time is shown in Figure 2. In all, there are eighteen FORTRAN statements, exclusive of the XEQ, LABEL, and END card. Each statement is punched on a separate IBM card. At times it may be necessary to change three cards in the main program depending on the data input, plus the input and output tape unit numbers. In order to obtain an understanding of when these changes should be made, a sample case will be discussed here. First, let us consider Card No. 3; the

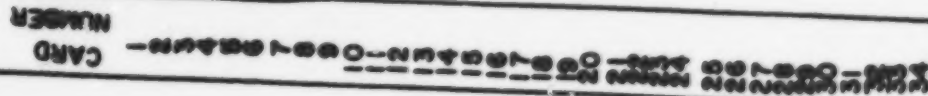
FORTTRAN statement appears as DIMENSION TIME (25), FIVE (25).

This instructs the computer to assign 25 storage locations for each variable TIME and FIVE. The FIVE written here is actual storage assigned for the D_i values. If more than 25 TIME values are to be used in calculating the dissolved oxygen deficits, it will be necessary to change both TIME (25) and FIVE (25) to read the number of TIME values desired. Let N represent the number of TIME values; then the statement in general form may be written DIMENSION TIME (N), FIVE (N). If the number N of desired TIME values is less than or equal to 25, no change to the statement is necessary.

Card No. 4 and Card No. 5 in the main program specify the input and output tape unit numbers used by the computing center. In preparing these cards, it is necessary to request the input and output tape unit numbers from the center where the program will be run. In general, the computing center will usually obtain the tape unit numbers; however, it is advisable to instruct them of the input-output numbers.

Card No. 4, appearing with the statement IT = 5, specifies that the input data tape unit number is 5. Similarly, KT = 6 on Card No. 5 specifies that the output tape unit number is 6.

Figure 3 shows one of the three subroutines used by the main program. Again, each statement appearing is punched on a separate IBM card. This subroutine is designed to aid in the



SUBROUTINE MAP USED BY THE MAIN PROGRAM

plotting of dissolved oxygen deficits as a function of time and is called into use by the main program by the statement appearing on Card No. 19. Figure 4 is the second subroutine that is used by the first to write out the computed data. The other subroutine is not shown here because of its existence on tape or card deck at the computing center. This type of subroutine is termed a library routine and is designed to perform calculations on the exponential function.

In punching the main program and subroutine on IBM cards, it is advisable to punch the main program on cards of a different color than the subroutine cards. This practice insures rapid identification when changes are to be made to the main program. Prior to actually running the program, a request to the keypunch operator for verification of the information punched on the cards is desirable.

IV. Preparation of Data

A sample case is shown in Table I of the input data. Here it is desired to compute and plot the dissolved oxygen deficits for three sets of data, each with a different initial oxygen deficit (D_a) and ultimate biochemical oxygen demand (L_a). It is assumed that $K_1 = 0.12$, and $K_2 = 0.25$ for each set. This is done here for reasons of simplicity; actually, K_1 and K_2 may take on different values for each set of data. Again, for simplicity, the desired deficits here are computed for each day

CARD NUMBER

```

CARD COLUMNS
SUBROUTINE WRITEM (B1,B2,B3,B4,B5,B6,X,FX,N,XMAX,XMIN,BLANK)
DIMENSION X(1),FX(1),B(6),OV(17)
100 FORMAT (1X,1PE10.3,2X,1H-,1X,17A6,1H-)
      B(1)=B1
      B(2)=B2
      B(3)=B3
      B(4)=B4
      B(5)=B5
      B(6)=B6
FACTOR=100.0/(XMAX-XMIN)
DO 2 K=1,17
    2 OV(K) = BLANK
DO 1 I=1,N
    JSCALE=IFX(I)-XMIN)*FACTOR+.5
M=XMODF(JSCALE,6)+1
L=JSCALE/6+1
OV(L)=B(M)
WRITE OUTPUT TAPE 6,100,X(1),(OV(K),K=1,17)
    1 OV(L) = BLANK
RETURN
END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)

```

SUBROUTINE WRITEM USED BY SUBROUTINE MAP

from 1 to 20 days for each set. The time values may be given in tenths or hundredths of a day, if desired; however, the interval between values should remain constant. In addition, for all practical purposes the program is designed to accept an unlimited and variable length number of time values from set to set. Refer to Figure 2 for the data setup.

Table I

<u>Set Number</u>	<u>D_a Initial Oxygen Deficit</u>	<u>Ultimate BOD</u>
1	0.00	5.00
2	1.00	10.00
3	2.00	15.00

To begin, the first data card will contain:

1. Number of time values N (In this case N = 20 for each set)
2. Set Number (Set No. 1)
3. Values for K_1 (deoxygenation rate constant)
4. Value for K_2 (reaeration rate constant)
5. Value of initial oxygen deficit (D_a)
6. Value of ultimate biochemical oxygen demand (L_a)

The second card will contain ten values of the time, starting with 1.00 and ending with 10.00. The third data card will start with 11.00 and end with 20.00. This completes the first set of data that the computer will work with. The second set of data will contain the six pieces of information contained on the first data card, only using the values that pertain to Set No. 2. Again, since our sample case specifies the same time

DATA PACKAGE ENDS WITH END CARD PUNCHED: column 7,8,9 punched END

SET 2,3,4 FOLLOW

ADDITIONAL TIME CARD CONTAINING VALUES 11.00-20.00

SET 1 DATA PACK

		TIME											
		1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00		
ROSET	1	0.12	0.25	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3333333333	4444444444
	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5555555555	6666666666
	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7777777777	8888888888
	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9999999999	0000000000
DO DEFICIT		0.00											
ULTIMATE DO		0.00											

FIRST DATA CARD

SECOND DATA CARD

DATA PACKAGE BEGINS WITH LEAD CARD PUNCHED: column 1 punched 8
7,8,9,10 punched DATA

Data package format for set number 1

7-a

FIGURE 5

values as the first set, the second and third cards of the second set will be identical to the second and third cards of the first set. Similarly, Set No. 3 will be punched in the order specified above. The number of sets is unlimited, since each set is read into memory prior to the calculations, and the second set of data will not be read in until the completed plot of the calculated information from the first set is finished. After the last set of data is worked on, the program halts.

In punching the data, the following format must be rigidly adhered to:

1. The number of time values N should be punched from right to left, beginning in Card Column 4; i.e., if nine values of the time are to be used, a 9 punch should appear in Column 4. If $N = 20$, as in our case here, the 0 appears in Column 4, and the 2 appears in Column 3.
2. The set number is to be punched in Column 10. Again, if the set number exceeds 9, the last digit appears in Column 10. Columns 5, 6, and 7 should contain the word SET.
3. The deoxygenation rate constant K_1 is punched in Columns 19 and 20, with the decimal punch appearing in Column 18. The decimal point must always appear in Column 18, whether the number is an integer or decimal.
4. The reaeration rate constant K_2 is punched in Columns 29 and 30, with the decimal point punched in Column 28. Again, the same holds regarding the decimal point punch as above.
5. Initial oxygen deficit is punched with the integer value in Column 37, decimal point in Column 38, and tenths and hundredths in Columns 39 and 40, respectively.

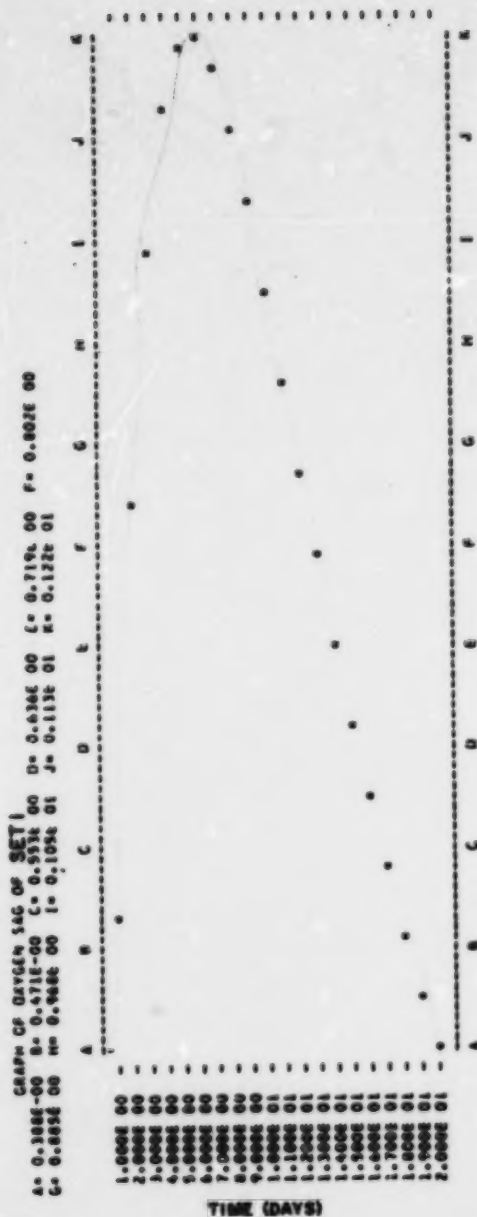
6. The ultimate biochemical oxygen demand is punched in Columns 46 through 50, with decimal punch in Column 48.

The time values begin on the second data card of each set, with the first appearing in Columns 1 through 7. The decimal point punch appears in Column 5. Decimal points may be located on each data card beginning with the second card by adding seven columns successively; i.e., decimal point for the second time value would be punched in Column 12, the third in Column 19, etc., until the tenth decimal point appears in Column 68.

V. Data Output

Figures 6, 7, and 8 show the output for our sample case using three sets of input data. First, the run identification number indicates the output data which follows is for a specific set number. The input values for time are printed, together with the computed oxygen deficits corresponding to a particular time value. The printed data for each set is then plotted directly.

A graph of the oxygen sag appears with the time values on the horizontal ordinate and oxygen deficit values along the vertical ordinate. The time values along the horizontal ordinate appear in floating point form where E 00 represents the exponent to which 10 is raised. In conversion to decimal form, it is necessary to multiply the number preceding the E by 10 to the appropriate power.

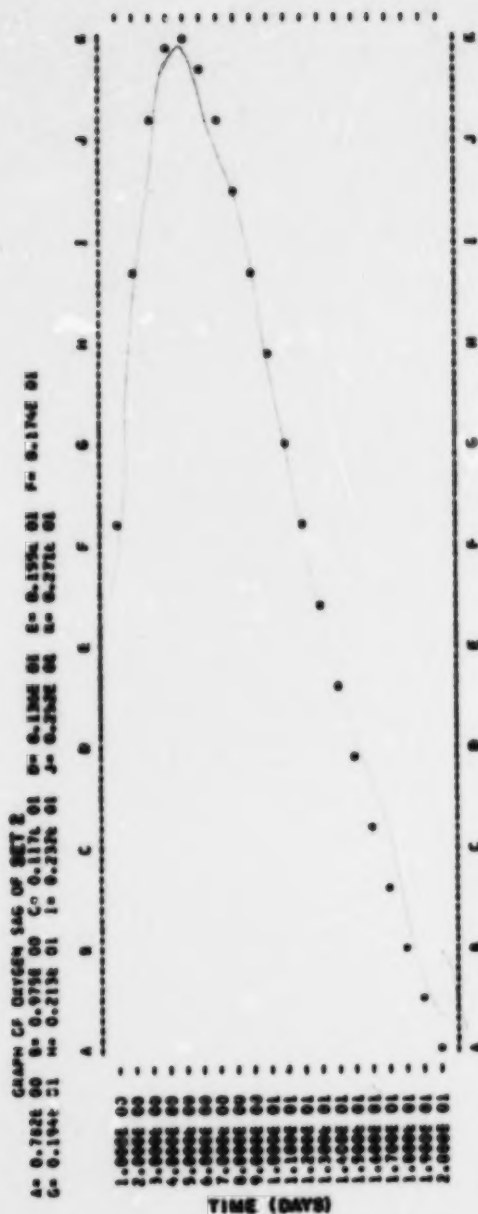


DATA OUTPUT AND GRAPH FOR SET NUMBER 1

RUN IDENTIFICATION SET 1		Set number	
TIME	OR. SAC	20 SET 1	Number of time values
1.00	0.49991	0.25	K_2 Value
2.00	0.53122	0.17	K_1 Value
3.00	1.03999		
4.00	1.15002		
5.00	1.21005		
6.00	1.21072		
7.00	1.19048		
8.00	1.14257		
9.00	1.00091		
10.00	1.01127		
11.00	0.93768		
12.00	0.86373		
13.00	0.79050		
14.00	0.72052		
15.00	0.65437		
16.00	0.59211		
17.00	0.53430		
18.00	0.48100		
19.00	0.43215		
20.00	0.38760		
	0.		
			Ultimate BOD
			Initial DO Deficit

ga

FIGURE 6



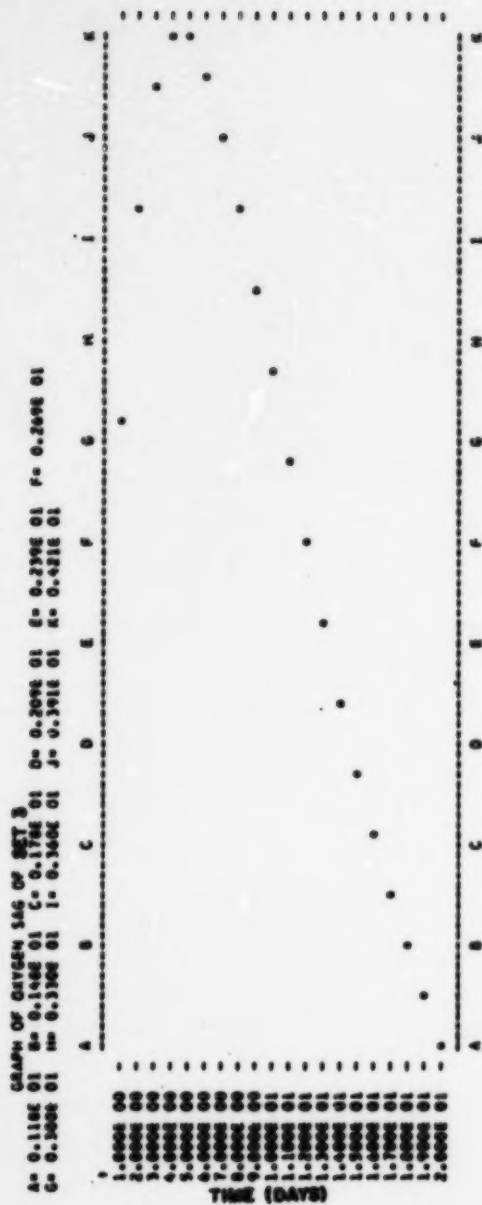
RUN IDENTIFICATION SET 2

TIME	O ₂ , SAG	Set number	Number of time values
1.00	1.77643	1.00	1.00
2.00	2.36497	2.00	2.00
3.00	2.55215	3.00	3.00
4.00	2.48392	4.00	4.00
5.00	2.79700	5.00	5.00
6.00	2.65056	6.00	6.00
7.00	2.59073	7.00	7.00
8.00	2.62748	8.00	8.00
9.00	2.68721	9.00	9.00
10.00	2.10163	10.00	10.00
11.00	1.93169	11.00	11.00
12.00	1.77356	12.00	12.00
13.00	1.62056	13.00	13.00
14.00	1.47185	14.00	14.00
15.00	1.33227	15.00	15.00
16.00	1.20294	16.00	16.00
17.00	1.08386	17.00	17.00
18.00	0.97310	18.00	18.00
19.00	0.87295	19.00	19.00
20.00	0.78194	20.00	20.00

DATA OUTPUT AND GRAPH FOR SET NUMBER 2

9-4

FIGURE 7



RUN IDENTIFICATION SET 3

TIME	OR, SAG
1.00	3.05444
2.00	3.78471
3.00	4.06441
4.00	4.20981
5.00	4.30495
6.00	4.35441
7.00	4.37508
8.00	4.38328
9.00	4.38352
10.00	4.37799
11.00	4.36150
12.00	4.33975
13.00	4.31275
14.00	4.28016
15.00	4.24197
16.00	4.19948
17.00	4.15321
18.00	4.10375
19.00	4.05128
20.00	3.99571

Ultimate BOD 15.00

Initial DO Deficit 2.60

K_2 Value 0.25

K_1 Value 0.12

Set number 3

Number of time values 20

DATA OUTPUT AND GRAPH FOR SET NUMBER 3

9-C

FIGURE 8

The vertical ordinate is developed by taking the largest value of the computed oxygen deficit and letting it be represented by the alphabetic character K, with the smallest computed oxygen deficit represented by the character A. The range of values for the deficit then becomes K-A. The alphabetic characters from A through K are assigned floating point values, as shown above each graph.

VI. Cost Data

For actual production runs, using the IBM 7090, the machine rental rates vary from approximately \$360 per hour to \$500 per hour. The rental time required to run the cases shown in Figures 6, 7, and 8 amounted to less than one minute, or approximately \$6 for a minimum machine rental. Since costs and methods of charging vary from one computer installation to the next, it is advisable to determine if a minimum rental charge exists. In other words, certain installations may require a minimum of five or ten minute charge regardless of actual machine time. In this case, it would be advisable to run many sets of data at one time or search for another installation. Since computer rates vary, it is usually advisable to shop; however, this should be justified by the rate differential, as the administrative time cost involved should not exceed the monies saved by procuring a lower rate computer charge. In addition, the procurement of a computer close to the technical personnel using

the machine will produce a savings far in excess of the differential between low and high computer rentals.

VII. Conclusions

The program package has been shown to consist of four separate card decks, namely:

1. Main Program Deck
2. Subroutine Deck Map
3. Subroutine Deck Written
4. Data Input Deck

Assuming the program will be re-run at the same computing center, no changes need be made in the symbolic decks. Hence, to conserve computing time, we may replace (1), (2), and (3) by the corresponding binary decks. (These are machine language programs which are compiled by the FORTRAN program during the first run.)

VIII. Acknowledgments

The authors wish to express their appreciation to Mr. Emanuel Mehr, Consulting Computer Mathematician, New York University, and to Mr. P. O'Hara, Mathematician, National Bureau of Standards, Washington, D. C., for their cooperation in developing this technique.